

# MaxHeap.java

```

1 /*
2  * Written by JJ Shepherd
3  */
4 public class MaxHeap <T extends Comparable<T>>
5 {
6     private T[] heap;
7     private int size;//First null element
8     public static final int DEF_SIZE = 128;
9
10    public MaxHeap()
11    {
12        init(DEF_SIZE);
13    }
14    public MaxHeap(int aSize)
15    {
16        init(aSize);
17    }
18    private void init(int aSize)
19    {
20        if(aSize <= 0)
21        {
22            init(DEF_SIZE);
23            return;
24        }
25        heap = (T[])(new Comparable[aSize]);
26        size = 0;
27    }
28    public void add(T aData)
29    {
30        if(heap[heap.length-1] != null)
31            return;
32        heap[size] = aData;
33        bubbleUp();
34        size++;
35    }
36    private void bubbleUp()
37    {
38        int index = size;
39        while(index > 0 && heap[(index-1)/2].compareTo(heap[index])<0)
40        {
41            T temp = heap[(index-1)/2];
42            heap[(index-1)/2] = heap[index];
43            heap[index] = temp;
44            index = (index-1)/2;
45        }
46    }
47    public T remove()
48    {
49        if(size <= 0)
50            return null;
51        T ret = heap[0];
52        heap[0] = heap[size-1];
53        heap[size-1] = null;
54        size--;
55        bubbleDown();
56        return ret;
57    }

```

```
58 private void bubbleDown()  
59 {  
60     int index = 0;  
61     while(index*2+1 < size)  
62     {  
63         int bigIndex = index*2+1;  
64         if(index*2+2 < size && heap[index*2+1].compareTo(heap[index*2+2])<0)  
65             bigIndex = index*2+2;  
66         if(heap[index].compareTo(heap[bigIndex])<0)  
67         {  
68             T temp = heap[index];  
69             heap[index] = heap[bigIndex];  
70             heap[bigIndex] = temp;  
71         }  
72         else  
73             break;  
74         index = bigIndex;  
75     }  
76 }  
77 public T peek()  
78 {  
79     return heap[0];  
80 }  
81 public void print()  
82 {  
83     for(int i=0;i<size;i++)  
84         System.out.println(heap[i]);  
85 }  
86 }  
87
```